UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/638,268	08/14/2000	Paul David Bryan	RAL9-2000-0069-US1	9608

25299 7590 10/14/2004

IBM CORPORATION
PO BOX 12195
DEPT 9CCA, BLDG 002
RESEARCH TRIANGLE PARK, NC 27709

EXAMINER	
THANGAVELU, KANDASAMY	
ART UNIT	**PAPER NUMBER**
2123	

DATE MAILED: 10/14/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/638,268	BRYAN ET AL.
	Examiner	Art Unit
	Kandasamy Thangavelu	2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 July 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *14 August 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Introduction

1. This communication is in response to the Applicants' Response mailed on July 29, 2004. Claims 1-21 of the application are pending. This office action is made non-final.

Drawings

2. The drawings are objected to; see a copy of Form PTO-948 sent with previous Office action, Paper No. 4.

Specification

3. The disclosure is objected to because of the following informalities:

Page 10, Lines 17-19, "In order to systematically enumerate each possible parameter combination, the generator assigns a significance level to each parameter combination specification in a command statement, based on its place in a sequence within the command statement" appears to be incorrect, as it contradicts with the process described in Fig. 3 and Page 10, Line 23 to Page 13, Line 23; and it appears that it should be "In order to systematically enumerate each possible parameter

combination, the generator assigns a significance level to each parameter specification in a command statement, based on its place in a sequence within the command statement".

Page 11, Lines 14-15 state, "as shown in block 308, if within the current significance level, the upper limit of a range has not been reached... the parameter value of the range is updated; so the significance level is associated with a parameter and not a parameter combination. Page 11, Lines 19-20 state, "when the upper limit of a range is reached ..., the generator moves up to the next significance level", which also indicates that the significance level is associated with a parameter. Specification, Page 10, Lines 20-22 need to be corrected to reflect this use of significance level.

Appropriate corrections are requested.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. §112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claim 21 is rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Claim 21 states in part, "the generator generates ***a transaction from each parameter*** of the command statement beginning with the lowest level of significance". There is no support for generating a transaction (bus transaction or a test case) from each parameter of the command statement in the specification. Therefore one of ordinary skill in the art will not know how to generate ***a transaction from each parameter***. A transaction involves a combination of parameters and specific values for the parameters. The claim appears to be incorrect. It is not clear as to how "beginning with the lowest level of significance" is used in generating the transaction if a transaction is generated for each parameter. It is also not clear as to what the lowest level of significance refers to in that case.

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 20 and 21 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 20 recites, "the command statement identifies a significance level with each parameter combination ". The significance level for each parameter combination as specified in Page 10, Lines 17-22 is incorrect as it contradicts with the process described in Fig. 3 and Page 10, Line 23 to Page 13, Line 23, as explained in

Paragraph 3 above; therefore, "identifies a significance level with each parameter combination" is vague and indefinite.

Claim 21 recites, "the generator generates ***a transaction from each parameter*** of the command statement beginning with the lowest level of significance". A transaction from each parameter is vague and indefinite since it is not described in the specification. It is also not clear as to what the lowest level of significance refers to.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

9. The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

10. Claims 1, 3-5, 7, 12 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Meyer** (U.S. Patent 6,571,204) in view of **Mongan** (U.S. Patent 6,378,088).

10.1 **Meyer** teaches bus modeling language generator. Specifically as per Claim 1, **Meyer** teaches providing a design-under-test (DUT) specification of bus transaction types and parameters corresponding to the DUT; and testing bus transactions for verification of the DUT (CL1, L12-21; CL1, L59-64; CL1, L65 to CL2, L7; CL2, L40-55; CL4, L41-50).

Meyer does not expressly teach providing a design-under-test (DUT) configuration file comprising a specification of bus transaction types and parameters corresponding to the DUT. **Mongan** teaches providing a design-under-test (DUT) configuration file (called description file) comprising a specification of actions (transaction types) and data (parameters) corresponding to the DUT (CL3, L56-63; CL5, L13-15; CL5, L28-29; CL5, L41-47), because the configuration file (called description file) is the heart of the test generator; all information that the test generator needs about the DUT is in the description file; and the tests generated by the test generator have all actions and data hard coded (CL5, L41-44; CL5, L13-15). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising a design-under-test (DUT) specification of bus transaction types and parameters corresponding to the DUT with the method of **Mongan** that included providing a design-under-test (DUT) configuration file (called description file) comprising a specification of actions (transaction types) and data (parameters) corresponding to the DUT. The artisan would have been motivated because the configuration file (called description file) would be the heart of

the test generator; all information that the test generator needed about the DUT would be in the description file; and the tests generated by the test generator would have all actions and data hard coded.

Meyer teaches testing bus transactions for verification of the DUT (CL1, L59-64; CL1, L65 to CL2, L7; CL4, L41-50). **Meyer** does not expressly teach processing the configuration file to generate a test case comprising bus transactions for verification of the DUT. **Mongan** teaches processing the configuration file (called description file) to generate a test case comprising various actions and data for verification of the DUT (CL1, L23-24; CL2, L47-52; CL3, L22-26; CL3, L52-55), because that allows generating tests consisting of random data and random series of actions (CL2, L50-52); so coverage of combinations of actions that is superior to an ad hoc test design is achieved and many fatal defects that would be missed by an ad hoc test design can be discovered; testing using this technique is more easily quantified; it lends itself to a high degree of automation making it less expensive than ad hoc testing (CL3, L4-7; CL3, L9-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising testing bus transactions for verification of the DUT with the method of **Mongan** that included processing the configuration file (called description file) to generate a test case comprising various actions and data for verification of the DUT. The artisan would have been motivated because that would allow generating tests consisting of random data and random series of actions; so coverage of combinations of actions that would be superior to an ad hoc test design would be achieved and many fatal defects that would be missed by an ad hoc test design could be discovered; testing using this technique would

be more easily quantified; it would lend itself to a high degree of automation making it less expensive than ad hoc testing.

10.2 As per Claim 3, **Meyer** and **Mongan** teach the method of claim 1. **Meyer** does not expressly teach that the processing step comprises converting the specification into a plurality of combinations of the parameters. **Mongan** teaches that the processing step comprises converting the specification into a plurality of combinations of the parameters (CL3, L56-63; CL5, L13-15), because the tests generated by the test generator have all actions and data hard coded (CL5, L13-15). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising a design-under-test (DUT) specification of bus transaction types and parameters corresponding to the DUT with the method of **Mongan** that included the processing step comprising converting the specification into a plurality of combinations of the parameters. The artisan would have been motivated because the tests generated by the test generator would have all actions and data hard coded.

Per Claim 4: **Meyer** teaches applying the bus transactions to the DUT for verification (CL4, L41-50; CL5, L10-18).

10.3 As per Claim 5, **Meyer** teaches a design-under-test (DUT); verification of the DUT (CL1, L59-64; CL4, L41-50); and possible parameter combinations for bus transactions of the DUT (CL1, L65 to CL2, L7; CL2, L40-55).

Meyer does not expressly teach describing a DUT in a configuration file using a condensed syntax. **Mongan** teaches describing a DUT in a configuration file (called description file) using a condensed syntax (CL3, L56-63; CL5, L41-67; CL6, L31-34), because the configuration file (called description file) is the heart of the test generator; all information that the test generator needs about the DUT is in the description file (CL5, L41-44). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising a design-under-test (DUT) with the method of **Mongan** that included describing a DUT in a configuration file using a condensed syntax. The artisan would have been motivated because the configuration file (called description file) would be the heart of the test generator; all information that the test generator needed about the DUT would be in the description file.

Meyer teaches testing possible parameter combinations for bus transactions of the DUT (CL1, L65 to CL2, L7; CL2, L40-55). **Meyer** does not expressly teach generating a test case for verification of the DUT by converting the condensed syntax into an enumeration of possible parameter combinations for bus transactions of the DUT. **Mongan** teaches generating a test case for verification of the DUT by converting the condensed syntax into an enumeration of possible parameter combinations for bus transactions of the DUT (CL1, L23-24; CL2, L47-52; CL3, L22-26; CL3, L52-55; CL5, L13-15; CL5, L41-67; CL6, L31-34), because that allows generating tests consisting of random data and random series of actions (CL2, L50-52); so coverage of combinations of actions that is superior to an ad hoc test design is achieved and many fatal defects that would be missed by an ad hoc test design can be discovered; testing using this technique is more easily quantified; it lends itself to a high degree of automation making it less

expensive than ad hoc testing (CL3, L4-7; CL3, L9-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising testing possible parameter combinations for bus transactions of the DUT with the method of **Mongan** that included generating a test case for verification of the DUT by converting the condensed syntax into an enumeration of possible parameter combinations for bus transactions of the DUT. The artisan would have been motivated because that would allow generating tests consisting of random data and random series of actions; so coverage of combinations of actions that would be superior to an ad hoc test design would be achieved and many fatal defects that would be missed by an ad hoc test design could be discovered; testing using this technique would be more easily quantified; it would lend itself to a high degree of automation making it less expensive than ad hoc testing.

10.4 As per Claim 7, **Meyer** and **Mongan** teach the method of claim 5. **Meyer** teaches a range of parameter values for the bus transactions (CL1, L65 to CL2, L7; CL2, L40-55).

Meyer does not expressly teach that the syntax specifies a range of parameter values for the bus transactions. **Mongan** teaches that the syntax specifies a range of parameter values (CL3, L56-63; CL5, L13-15; CL5, L41-47), because the tests generated by the test generator have all actions and data hard coded (CL5, L13-15). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising a range of parameter values for the bus transactions with the method of **Mongan** that included the syntax specifying a range of parameter values. The artisan would have been

motivated because the tests generated by the test generator would have all actions and data hard coded.

10.5 As per Claim 12, **Meyer** teaches computer-usable medium storing computer-executable instructions, the instructions when executed implementing a process (Fig. 1, Item 113); comprising

evaluating a DUT defining transaction types and parameters corresponding to the DUT (CL1, L12-21; CL1, L59-64; CL1, L65 to CL2, L7; CL2, L40-55; CL4, L41-50).

Meyer does not expressly teach evaluating a syntax of a DUT configuration file; and generating bus functional language statements from the syntax. **Mongan** teaches evaluating a syntax of a DUT configuration file; and generating third party language statements from the syntax (CL3, L56-63; CL5, L41-67; CL6, L31-34; CL5, L13-15), because the configuration file (called description file) is the heart of the test generator; and the tests generated by the test generator are scripts that have all actions and data hard coded for execution by a third party tool (CL5, L41-44; CL5, L13-15). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the computer-usable medium of **Meyer** comprising evaluating a syntax of a DUT configuration file; and generating third party language statements from the syntax with the computer-usable medium of **Mongan** that included evaluating a syntax of a DUT configuration file; and generating third party language statements from the syntax. The artisan would have been motivated because the configuration file (called description file) would

be the heart of the test generator; and the tests generated by the test generator would be scripts that would have all actions and data hard coded for execution by a third party tool.

Meyer teaches that the BFM receive a bus function language program that controls the bus signals sent between the BFM and the DUT (CL5, L10-13). **Mongan** teaches that the tests generated by the test generator are scripts with all actions and data hard coded so they could be executed by a third party testing tool (CL5, L13-15). Therefore, it would have been obvious to one of ordinary skill in the art to modify the computer-usable medium of **Meyer** with the computer-usable medium of **Mongan** and then change the output language for generating bus functional language statements (CL5, L10-28; Fig. 3; CL5, L40-50). The artisan would have been motivated because the BFM models used by **Meyer** require bus functional language statements (CL5, L10-28; Fig. 3; CL5, L40-50).

10.6 As per Claim 15, **Meyer** teaches a system (Fig. 1); comprising

a memory including computer-executable instructions (Fig. 1, Item 105);

a processor coupled to the memory for executing the instructions (Fig. 1, Item 101); and

bus transaction types and parameters corresponding to the DUT; and bus transactions for verification of the DUT (CL1, L12-21; CL1, L59-64; CL1, L65 to CL2, L7; CL2, L40-55; CL4, L41-50).

Meyer does not expressly teach a configuration file for a DUT including bus transaction types and parameters corresponding to the DUT. **Mongan** teaches a configuration file (called description file) for a DUT comprising a specification of actions (transaction types) and data

(parameters) corresponding to the DUT (CL3, L56-63; CL5, L13-15; CL5, L28-29; CL5, L41-47), because the configuration file (called description file) is the heart of the test generator; all information that the test generator needs about the DUT is in the description file; and the tests generated by the test generator have all actions and data hard coded (CL5, L41-44; CL5, L13-15). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the system of **Meyer** comprising a DUT including bus transaction types and parameters corresponding to the DUT with the system of **Mongan** that included a configuration file (called description file) for a DUT comprising a specification of actions (transaction types) and data (parameters) corresponding to the DUT. The artisan would have been motivated because the configuration file (called description file) would be the heart of the test generator; all information that the test generator needed about the DUT would be in the description file; and the tests generated by the test generator would have all actions and data hard coded.

Meyer teaches bus transactions for verification of the DUT (CL1, L59-64; CL1, L65 to CL2, L7; CL4, L41-50). **Meyer** does not expressly teach that the instructions process the configuration file to generate bus transactions for verification of the DUT. **Mongan** teaches that the instructions process the configuration file (called description file) to generate a test case comprising various actions and data for verification of the DUT (CL1, L23-24; CL2, L47-52; CL3, L22-26; CL3, L52-55), because that allows generating tests consisting of random data and random series of actions (CL2, L50-52); so coverage of combinations of actions that is superior to an ad hoc test design is achieved and many fatal defects that would be missed by an ad hoc test design can be discovered; testing using this technique is more easily quantified; it lends itself

to a high degree of automation making it less expensive than ad hoc testing (CL3, L4-7; CL3, L9-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the system of **Meyer** comprising bus transactions for verification of the DUT with the system of **Mongan** that included the instructions processing the configuration file (called description file) to generate a test case comprising various actions and data for verification of the DUT. The artisan would have been motivated because that would allow generating tests consisting of random data and random series of actions; so coverage of combinations of actions that would be superior to an ad hoc test design would be achieved and many fatal defects that would be missed by an ad hoc test design could be discovered; testing using this technique would be more easily quantified; it would lend itself to a high degree of automation making it less expensive than ad hoc testing.

11. Claims 2, 6, 8, 10, 11, 13, 14 and 16-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Meyer** (U.S. Patent 6,571,204) in view of **Mongan** (U.S. Patent 6,378,088), and further in view of **Shrote** (U.S. Patent 5,774,358).

11.1 As per Claim 2, **Meyer** and **Mongan** teach the method of claim 1. **Meyer** teaches bus transaction types and testing bus transactions for verification of the DUT (CL1, L65 to CL2, L7; CL2, L40-55). **Meyer** does not expressly teach that the processing step further comprises evaluating rules in the configuration file to include or exclude selected ones of the bus transactions from the test case. **Shrote** teaches that the processing step further comprises evaluating rules in the configuration file to include or exclude selected ones of the system

elements from the test case (CL8, L19-33), as that allows generating the test instructions and data based on the particular rules provided by the user to meet the requirements of the test case (CL8, L26-28). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Meyer** with the method of **Shrote** that included the processing step further comprising evaluating rules in the configuration file to include or exclude selected ones of the system elements from the test case. The artisan would have been motivated because that would allow generating the bus transaction test instructions and data based on the particular rules provided by the user to meet the requirements of the test case.

11.2 As per Claim 6, **Meyer** and **Mongan** teach the method of claim 5. **Meyer** does not expressly teach including rules in the configuration file to include or exclude parameter combinations from the enumeration. **Shrote** teaches including rules in the configuration file to include or exclude parameter combinations from the enumeration (CL8, L19-33), as that allows generating the test instructions and data based on the particular rules provided by the user to meet the requirements of the test case (CL8, L26-28). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Meyer** with the method of **Shrote** that included including rules in the configuration file to include or exclude parameter combinations from the enumeration. The artisan would have been motivated because that would allow generating the bus transaction test instructions and data based on the particular rules provided by the user to meet the requirements of the test case.

11.3 As per Claim 8, **Meyer** and **Mongan** teach the method of claim 5. **Meyer** teaches transaction types and a set of parameters for each transaction type (CL1, L65 to CL2, L7; CL2, L40-55).

Meyer does not expressly teach that the syntax specifies transaction types, a set of parameters for each transaction type. **Mongan** teaches that the syntax specifies transaction types (actions), a set of parameters (data) for each transaction type (CL3, L56-63; CL5, L13-15; CL5, L41-47), because the tests generated by the test generator have all actions and data hard coded (CL5, L13-15). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising transaction types and a set of parameters for each transaction type with the method of **Mongan** that included the syntax specifying transaction types, a set of parameters for each transaction type. The artisan would have been motivated because the tests generated by the test generator would have all actions and data hard coded.

Meyer does not expressly teach directives for determining a mode of the converting. **Shrote** teaches directives for determining a mode of the converting (Fig. 3A, Items 308 and 314; CL12, L37-43), as directives are input for particular tests to set the boundaries of testing and generate appropriate instruction/data stream for that test (CL12, L37-43). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** with the method of **Shrote** that included directives for determining a mode of the converting, as directives would be input for particular tests to set the boundaries of testing and generate appropriate instruction/data stream for that test.

11.4 As per Claims 10 and 11, **Meyer, Mongan** and **Shrote** teach the method of claim 8. **Meyer** does not expressly teach that the directives cause values for the parameters to be evaluated as a list; and the directives cause the transaction types to be selected at random. **Mongan** teaches that the directives cause values for the parameters to be evaluated as a list (CL5, L13-15); and the directives cause the transaction types to be selected at random (CL2, L49-52), because that allows generating tests consisting of random data and random series of actions (CL2, L50-52); so coverage of combinations of actions that is superior to an ad hoc test design is achieved and many fatal defects that would be missed by an ad hoc test design can be discovered; testing using this technique is more easily quantified; it lends itself to a high degree of automation making it less expensive than ad hoc testing (CL3, L4-7; CL3, L9-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Meyer** comprising testing bus transactions for verification of the DUT with the method of **Mongan** that included the directives causing values for the parameters to be evaluated as a list; and the directives causing the transaction types to be selected at random. The artisan would have been motivated because that would allow generating tests consisting of random data and random series of actions; so coverage of combinations of actions that would be superior to an ad hoc test design would be achieved and many fatal defects that would be missed by an ad hoc test design could be discovered; testing using this technique would be more easily quantified; it would lend itself to a high degree of automation making it less expensive than ad hoc testing.

11.5 As per Claim 13, **Meyer** and **Mongan** teach the computer-usable medium of claim 12. **Meyer** teaches selected bus functional language statements (CL5, L10-28; Fig. 3; CL5, L40-50).

Meyer does not expressly teach that the configuration file further includes rules for including or excluding selected bus functional language statements from being generated. **Shrote** teaches that the configuration file further includes rules for including or excluding selected bus functional language statements from being generated (CL8, L19-33), as that allows generating the test instructions and data based on the particular rules provided by the user to meet the requirements of the test case (CL8, L26-28). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the computer-usable medium of **Meyer** with the computer-usable medium of **Shrote** that included the configuration file further including rules for including or excluding selected bus functional language statements from being generated. The artisan would have been motivated because that would allow generating the bus transaction test instructions and data based on the particular rules provided by the user to meet the requirements of the test case.

11.6 As per Claim 14, **Meyer** and **Mongan** teach the computer-usable medium of claim 12. **Meyer** teaches outputting the parameter combination in a bus functional language statement (CL5, L10-28; Fig. 3; CL5, L40-50).

Meyer does not expressly teach testing a parameter combination generated from the configuration file against the rules; and outputting the parameter combination in a bus functional language statement when the parameter combination is not excluded by the rules. **Shrote** teaches testing a parameter combination generated from the configuration file against the rules;

and outputting the parameter combination when the parameter combination is not excluded by the rules (CL8, L19-33), as that allows generating the test instructions and data based on the particular rules provided by the user to meet the requirements of the test case (CL8, L26-28). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the computer-usable medium of **Meyer** with the computer-usable medium of **Shrote** that included testing a parameter combination generated from the configuration file against the rules; and outputting the parameter combination when the parameter combination is not excluded by the rules. The artisan would have been motivated because that would allow generating the bus transaction test instructions and data based on the particular rules provided by the user to meet the requirements of the test case.

11.7 As per Claim 16, **Meyer** and **Mongan** teach the system of claim 15. **Meyer** does not expressly teach that the configuration file includes rules for including or excluding selected bus transactions from being generated. **Shrote** teaches that the configuration file includes rules for including or excluding selected bus transactions from being generated (CL8, L19-33), as that allows generating the test instructions and data based on the particular rules provided by the user to meet the requirements of the test case (CL8, L26-28). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the system of **Meyer** with the system of **Shrote** that included the configuration file including rules for including or excluding selected bus transactions from being generated. The artisan would have been motivated because that would allow generating the bus transaction test instructions and data based on the particular rules provided by the user to meet the requirements of the test case.

11.8 As per Claim 17, **Meyer** teaches preparing specifications of parameter combinations corresponding to buss transactions of a device under test (CL1, L65 to CL2, L7; CL2, L40-55; CL1, L59-64; CL4, L41-50).

Meyer does not expressly teach a method for generating a test case for a buss interface. **Mongan** teaches a method for generating a test case for a buss interface (CL1, L23-24; CL2, L47-52; CL3, L22-26; CL3, L52-55), because that allows generating tests consisting of random data and random series of actions (CL2, L50-52); so coverage of combinations of actions that is superior to an ad hoc test design is achieved and many fatal defects that would be missed by an ad hoc test design can be discovered; testing using this technique is more easily quantified; it lends itself to a high degree of automation making it less expensive than ad hoc testing (CL3, L4-7; CL3, L9-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising preparing specifications of parameter combinations corresponding to buss transactions of a device under test with the method of **Mongan** that included a method for generating a test case for a buss interface. The artisan would have been motivated because that would allow generating tests consisting of random data and random series of actions; so coverage of combinations of actions that would be superior to an ad hoc test design would be achieved and many fatal defects that would be missed by an ad hoc test design could be discovered; testing using this technique would be more easily quantified; it would lend itself to a high degree of automation making it less expensive than ad hoc testing.

Meyer does not expressly teach forming a configuration file of the parameter combinations in a condensed syntax including commands. **Mongan** teaches forming a configuration file (called description file) of the parameter combinations in a condensed syntax including commands (CL3, L56-63; CL5, L41-67; CL6, L31-34), because the configuration file (called description file) is the heart of the test generator; all information that the test generator needs about the DUT is in the description file (CL5, L41-44). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** with the method of **Mongan** that included describing forming a configuration file (called description file) of the parameter combinations in a condensed syntax including commands. The artisan would have been motivated because the configuration file (called description file) would be the heart of the test generator; all information that the test generator needed about the DUT would be in the description file.

Meyer does not expressly teach forming a configuration file including rules to select various parameter combinations to be included in or excluded from the test case. **Shrote** teaches forming a configuration file including rules to select various parameter combinations to be included in or excluded from the test case (CL8, L19-33), as that allows generating the test instructions and data based on the particular rules provided by the user to meet the requirements of the test case (CL8, L26-28). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Meyer** with the method of **Shrote** that included forming a configuration file including rules to select various parameter combinations to be included in or excluded from the test case. The artisan would have been motivated because

that would allow generating the bus transaction test instructions and data based on the particular rules provided by the user to meet the requirements of the test case.

Meyer does not expressly teach generating from the configuration file all bus transactions defined by the rules comprising the test case. **Mongan** teaches generating from the configuration file all actions defined by the rules comprising the test case (CL1, L23-24; CL2, L47-52; CL3, L22-26; CL3, L52-55), because that allows generating tests consisting of random data and random series of actions (CL2, L50-52); so coverage of combinations of actions that is superior to an ad hoc test design is achieved and many fatal defects that would be missed by an ad hoc test design can be discovered; testing using this technique is more easily quantified; it lends itself to a high degree of automation making it less expensive than ad hoc testing (CL3, L4-7; CL3, L9-12). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising bus transactions for verification of the DUT with the method of **Mongan** that included generating from the configuration file all actions defined by the rules comprising the test case. The artisan would have been motivated because that would allow generating tests consisting of random data and random series of actions; so coverage of combinations of actions that would be superior to an ad hoc test design would be achieved and many fatal defects that would be missed by an ad hoc test design could be discovered; testing using this technique would be more easily quantified; it would lend itself to a high degree of automation making it less expensive than ad hoc testing.

Meyer does not expressly teach storing the bus transactions in an output file for use in a bus simulator. **Shrote** teaches storing the bus transactions in an output file for use in a bus simulator (Fig 3B, Item 332; CL14, L25-27), as the output file can be used to verify bus

transactions in a simulation (CL14, L30-32). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Meyer** with the method of **Shrote** that included storing the bus transactions in an output file for use in a bus simulator, as the output file could be used to verify bus transactions in a simulation.

11.9 As per Claim 18, **Meyer, Mongan** and **Shrote** teach the method of claim 17. **Meyer** teaches defining transaction types to be generated and the parameters associated with each transaction type (CL1, L65 to CL2, L7; CL2, L40-55; CL1, L59-64; CL4, L41-50).

Meyer does not expressly teach the configuration file includes statements defining transaction types to be generated and command statements which specify the parameters associated with each transaction type. **Mongan** teaches the configuration file includes statements defining transaction types to be generated and command statements which specify the parameters associated with each transaction type (CL3, L56-63; CL5, L13-15; CL5, L28-29; CL5, L41-47), because the configuration file (called description file) is the heart of the test generator; all information that the test generator needs about the DUT is in the description file; and the tests generated by the test generator have all actions and data hard coded (CL5, L41-44; CL5, L13-15). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** comprising defining transaction types to be generated and the parameters associated with each transaction type with the method of **Mongan** that included the configuration file including statements defining transaction types to be generated and command statements which specified the parameters associated with each

transaction type. The artisan would have been motivated because the configuration file (called description file) would be the heart of the test generator; all information that the test generator needed about the DUT would be in the description file; and the tests generated by the test generator would have all actions and data hard coded.

11.10 As per Claim 19, **Meyer, Mongan** and **Shrote** teach the method of claim 18. **Meyer** does not expressly teach that the command identifies a subset of the parameters which limits the number of transactions in the test case. **Mongan** teaches that the command identifies a subset of the parameters which limits the number of transactions in the test case (CL3, L56-63; CL5, L13-15; CL5, L28-29; CL5, L41-47), because the tests generated by the test generator have all actions and data hard coded (CL5, L41-44; CL5, L13-15). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to combine the method of **Meyer** with the method of **Mongan** that included the command identifying a subset of the parameters which limited the number of transactions in the test case. The artisan would have been motivated because the tests generated by the test generator would have all actions and data hard coded.

12. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Meyer** (U.S. Patent 6,571,204) in view of **Mongan** (U.S. Patent 6,378,088), and further in view of **Shrote** (U.S. Patent 5,774,358) and **Mantooth et al.** (U.S. Patent 6,236,956).

12.1 As per Claim 9, **Meyer, Mongan** and **Shrote** teach the method of claim 8. **Meyer** does not expressly teach that the directives cause a value for the parameters to be stepwise

incremented. **Mantooth et al.** teaches that the directives cause a value for the parameters to be stepwise incremented (CL10, L2-38; CL23, L40-67), as that allows predicting the effects of parameter variations on the system by simulation by sweeping the parameters within a range in a series of steps in which the specified parameter value is incremented by a predetermined increment until the specified range is completed (CL23, L44-52). It would have been obvious to one of ordinary skill in the art at the time of Applicants' invention to modify the method of **Meyer** with the method of **Mantooth et al.** that included the directives causing a value for the parameters to be stepwise incremented, as that would allow predicting the effects of parameter variations on the system by simulation by sweeping the parameters within a range in a series of steps in which the specified parameter value is incremented by a predetermined increment until the specified range is completed.

Response to Arguments

13. Applicants' amendments filed on July 29, 2004 have been fully considered. Applicants' arguments regarding claim rejections under 35 USC 112 First Paragraph and Second Paragraph are not persuasive. Claim rejections under 35 USC 103 (a) using additional prior art are included in this office action.

13.1 As per the applicants' argument that "the Hellestrand reference fails to disclose any design under test configuration file as set forth in claim 1; … the Apostol, Jr. et al. reference does not disclose any system for generating test cases for a design under test; there are no

configuration files disclosed which would constitute a specification of bus transaction type and any parameter corresponding to the design under test; ... the Sheafor et al. reference does not disclose a design under test configuration file as claimed in the application; ... the concept of a configuration file for generating a test case comprising bus transactions to verify a design under test is totally absent from the cited reference; in the Shrote reference, there is no reference to a configuration file which is used to create test cases comprising bus transactions", the Examiner has used new references **Meyer** and **Mongan.**

Meyer teaches providing a design-under-test (DUT) specification of bus transaction types and parameters corresponding to the DUT; and testing bus transactions for verification of the DUT (CL1, L12-21; CL1, L59-64; CL1, L65 to CL2, L7; CL2, L40-55; CL4, L41-50).

Mongan teaches providing a design-under-test (DUT) configuration file (called description file) comprising a specification of actions (transaction types) and data (parameters) corresponding to the DUT (CL3, L56-63; CL5, L13-15; CL5, L28-29; CL5, L41-47), because the configuration file (called description file) is the heart of the test generator; all information that the test generator needs about the DUT is in the description file; and the tests generated by the test generator have all actions and data hard coded (CL5, L41-44; CL5, L13-15). **Mongan** teaches processing the configuration file (called description file) to generate a test case comprising various actions and data for verification of the DUT (CL1, L23-24; CL2, L47-52; CL3, L22-26; CL3, L52-55), because that allows generating tests consisting of random data and random series of actions (CL2, L50-52); so coverage of combinations of actions that is superior to an ad hoc test design is achieved and many fatal defects that would be missed by an ad hoc test design can be discovered; testing using this technique is more easily quantified; it lends itself

to a high degree of automation making it less expensive than ad hoc testing (CL3, L4-7; CL3, L9-12).

In addition, applicants' attention is also directed to the following references which use ***configuration files*** to ***specify the parameters*** that are used in ***automatic generation of test cases*** using automated test generators.

Hellestrand (U. S. Patent 6,263,302) uses cache simulator to simulate several types of cache structures using generic cache model that uses a **list of parameters** to describe the structure of the cache and **policies** governing its operations. A cache **configuration file** is used to specify the values of the parameters for a particular cache model. The cache configuration is defined by a list of physical structure parameters specified in the cache configuration description file having a specified syntax (CL14, L57-66).

Slutz (U.S. Patent 6,138,112) uses a **test generator** to produce a set of query language statements comprised of **randomly chosen elements** for testing one or more database management systems. A **configuration file** specifies **parameters of the test statements**, in terms of maximum elements, weights of different elements etc. (Abstract). Automated testing of database management systems is taught (CL1, L6-7). It speeds up **generation of database test cases** by orders of magnitude (CL2, L2-3). Reading the configuration data containing a set of test parameters and then constructing a number of test statements are used (CL2, L17-19). Fig.4 comprising Figs. 4A-4F shows the configuration files used in the process (Fig.3, Item 310; CL2. L43-44).

Enokido et al. (U. S. Patent 6,243,835) a test specification generation system; a data analysis device reads the statements written in a test configuration file (abstract). A test specification generation system uses a test configuration storage means for storing configuration file describing the fundamental configuration of the test specification and a test item generation means for generating a test item from the design information read by the design information reading means (CL2, L5-16).

13.2 As per the applicants' argument that "the concept of having rules to include or exclude selected ones of bus transactions from the test case is not suggested in any of the references", the examiner respectfully disagrees.

Shrote teaches having rules in the configuration file to include or exclude selected ones of the system elements from the test case (CL8, L19-33), as that allows generating the test instructions and data based on the particular rules provided by the user to meet the requirements of the test case (CL8, L26-28).

13.3 As per the applicants' argument that "each of the references fails to disclose any test case which is derived by converting the condensed syntax of the configuration file to a test case", the examiner respectfully disagrees.

Mongan teaches describing a DUT in a configuration file (called description file) using a condensed syntax (CL3, L56-63; CL5, L41-67; CL6, L31-34). **Mongan** teaches generating a test case for verification of the DUT by converting the condensed syntax into an enumeration of

possible parameter combinations for bus transactions of the DUT (CL1, L23-24; CL2, L47-52; CL3, L22-26; CL3, L52-55; CL5, L13-15; CL5, L41-67; CL6, L31-34).

13.4 As per the applicants' argument that "none of the references provide any disclosure for generating bus function language statements from the syntax", the examiner respectfully disagrees.

Meyer teaches that the BFM receive a bus function language program that controls the bus signals sent between the BFM and the DUT (CL5, L10-13). **Mongan** teaches that the tests generated by the test generator are scripts with all actions and data hard coded so they could be executed by a third party testing tool (CL5, L13-15). Therefore, it would have been obvious to one of ordinary skill in the art to modify the computer-usable medium of **Meyer** with the computer-usable medium of **Mongan** and then change the output language for generating bus functional language statements (CL5, L10-28; Fig. 3; CL5, L40-50). The artisan would have been motivated because the BFM models used by **Meyer** require bus functional language statements (CL5, L10-28; Fig. 3; CL5, L40-50).

Conclusion

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dr. Kandasamy Thangavelu whose telephone number is 703-305-0043, till October 27, 2004 and 571-272-3717 after October 27, 2004. The examiner can normally be reached on Monday through Friday from 8:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kevin Teska, can be reached on (703) 305-9704, till October 27, 2004 and 571-272-3716 after October 27, 2004. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-9600.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

K. Thangavelu
Art Unit 2123
October 3, 2004

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER